

POȘIBILITĂȚI DE MODELARE SPAȚIALĂ ÎN MEDIU PROGRAMAT

Magyari-Sáska Zsolt¹, Dr. Haidu Ionel²

Key words: programming environment, Delphi, macro, inovaGIS, IDRISI, spatial analysis

ABSTRACT

Performing a GIS analysis may require a repeated execution of some more complex algorithms. These can be done in various modes depending on the used GIS software. This article shows some of the existing possibilities, including the native ways offered by several GIS software (macros, script languages), but also shows some other ways using special function libraries and software development platforms.

The second category was investigated because, the traditionally existing modalities to construct and reuse an algorithm lack either on controlling issues (no cycle, no decision possibilities), or in interface construction. We don't want to recreate some GIS environment from bases, but to use the existing ones, in a general programming environment.

As an example, we resolved the same tutorial problem using all major possibilities based on IDRISI Kilimajaro Edition. The finally created GIS platform offers free parameterization and execution controlling combined with an easy interface.

1. Introducere

Modelarea spațială este o posibilitatea cheie oferită de Sistemele Geoinformatic. Efortul necesar pentru realizarea modelării spațiale este direct proporțională cu complexitatea problemei care se dorește modelată.

Pentru un domeniu și o zonă studiată, modelarea spațială înseamnă determinarea elementelor (variabilelor) care trebuie luate în considerare, utilizarea anumitor metode și procedee de calcul pentru a determina parametrii modelului, după care se poate construi modelul propriu zis.

Deseori, modalitatea de calcul poate fi utilizată și în alte situații ținând cont de caracteristicile specifice zonei studiate.

Definirea unui model spațial este deci echivalentă cu definirea următoarelor elemente:

- elemente de intrare, parametrii modelului
- pașii de execuție necesare determinării caracteristicilor modelului (valoare parametrilor)
- pașii de execuție pentru realizarea modelului propriu zis

Comenzile executate, natura lor, numărul de execuție, parametrii utilizați depind de cazuri concrete, sunt operații care necesită o flexibilitate sporită și o intervenție permanentă din parte celui care dorește să realizeze modelul.

Totodată realizarea modelului poate necesita parcurgerea repetată a anumitor pași, eventual cu valori diferite. În acest sens ar fi necesară reexecutarea comenzilor, aceasta însemnând realegerea aceluiași valori, dintre care doar puține la număr se vor schimba.

¹ Universitatea "Babeș-Bolyai" Cluj-Napoca, Facultatea de Geografie, Extensia Universitară Gheorgheni

² Universitatea "Babeș-Bolyai" Cluj-Napoca, Facultatea de Geografie

Apare necesitatea realizării unui model de calcul, prin care analizele complexe:

- pot fi reexecutate cu schimbarea ușoară a parametrilor modificați, fără a fi nevoie de respecificare tuturor valorilor
- se pot lua decizii pe parcursul analizei, unele fiind decizii fiind codificate prin algoritmi, dar și decizii externe (umane), posibilitate numită și control al fluxului

Proape fiecare Sistem Geoinformatic, oferă una sau mai multe posibilități de automatizare a analizelor. Aceste posibilități pot fi grupate în următoarele categorii:

- interne, care sunt strâns legate de mediul GIS, funcționând doar în interiorul acelui sistem, și de obicei nu oferă caracteristicile unui limbaj de programare de nivel înalt, cu toate caracteristicile sale cum ar fi: orientare spre obiecte, structuri de control (decizii, cicluri), recursivitate, interoperabilitate. În această categorie se integrează următoarele posibilități: macroui, limbaje script, medii grafice.
- cooperative, noțiune care înseamnă o conlucrare a mediului GIS cu un limbaj de programare ori prin faptul că acel limbaj este utilizabil în GIS, sau funcționalitatea GIS-ului este apelabil în limbajul de programare. În această categorie se integrează limbajele de programare integrate (ex. Avenue, Python), sau limbaje de programare de nivel înalt care pot integra funcționalitate unor SIG (ex. IDRISI API).
- externe, ceea ce înseamnă posibilități de sine stătătoare care pot accesa datele unor SIG, neavând de obicei posibilități de analiză avansată predefinită asupra acestor date (ex. inovaGIS)

Oricare dintre cele trei categorii mai sus amintite, poate fi considerată ca un mediu programat, din moment ce operații specificate sunt executate unul după altul fără intervenția obligatorie a omului.

2. Sisteme GIS analizate

Pentru studiul prezentat s-au analizat trei programe GIS cunoscute: IDRISI Kilimanjaro Edition, ERDAS IMAGINE 8.4, ArcView 3.2.

2.1 IDRISI Kilimanjaro Edition

Posibilități de modelare internă

- modelare grafică
- modelare pe bază de macroui

Posibilități de modelare cooperativă

- IDRISI API

Posibilități de modelare externă

- inovaGIS

IDRISI Kilimanjaro Edition este sistemul cu cele mai multe posibilități de modelare existent, dintre cele trei sisteme. Acest lucru se bazează pe modularitatea specifică sistemului, în care fiecare operație este de fapt un program de sine stătător.

Începând cu versiunea Kilimanjaro Edition, conține modelul de modelare grafică prin meniul: *Modeling-Macro Modeler*. Posibilitățile modelării grafice sunt limitate, prin faptul că nu se pot lua decizii pe parcursul analizei, partea de control al

fluxului fiind aproape inexistentă. Pot fi definite cicluri de execuție, însă cu număr de repetare predefinit, și pot fi create submodele, asemenea unor subprograme, din domeniul programării.

Cu toate neajunsurile, modelarea grafică în IDRISI este o posibilitate utilă, în care schimbarea de parametri pentru funcțiile alese se realizează ușor și interactiv. De remarcat că la anumite funcții nu se poate folosi toată gama de opțiuni existentă. Un alt impediment în utilizarea facilă a modelării grafice, este utilizarea unor fișiere text externe, care conțin anumite parametri al unor operații (ex. RECLASS).

Un avantaj al modelării grafice în IDRISI este posibilitatea de vizualizare imediată a oricărui rezultat.

Modelarea bazată pe macroui exista chiar și în primele versiuni ale sistemului. Pentru aceasta se realizează un fișier text, care va conține instrucțiunile (comenzile) care vor fi executate în ordine. Pentru fiecare comandă este specificată formatul macro, cu toți parametri necesari. În această variantă nu există nici o posibilitate de control al fluxului, însă orice comandă (operație) poate fi utilizată la adevărata lui valoare, orice parametru, orice posibilitate fiind accesibilă. Un avantaj al fișierelor macro este posibilitatea de parametrizare a lor, ceea ce oferă utilitate extinsă acestor modele. Rularea macroului creat se realizează prin meniul: *Modeling-Run Macro*.

Există mai multe posibilități de modelare prin limbaje de programare externe. Una dintre ele se bazează pe *IDRISI API*. Sistemul este astfel conceput, încât executabilul sistemului este și un server de aplicație bazată pe tehnologia COM, prin care se pot accesa diferitele module (operații) ale sale. Acest server poate fi înglobat în mai multe limbaje de programare de nivel înalt (Visual C++, Visual Basic, Borland Delphi, Borland C++ Builder, etc.). Prin această modalitate pe lângă utilizarea directă și nemijlocită a operațiilor din IDRISI, se oferă acces și la alte elemente caracteristice sistemului (setarea directorului de lucru, acces individual la valori din fișierele de descriere, acces la starea și la progresul de execuție a unei comenzi lansate, etc.). Modalitatea necesită existența sistemului pe calculator, și în momentul lansării aplicație (VB, VC++, Delphi, etc.) se lansează automat și sistemul IDRISI (dacă nu era lansat în prealabil).

Prin IDRISI API, avem posibilitatea de a crea o interfață de lucru specifică domeniului studiat, aici se pot modifica parametri comenzilor, putându-se verifica corectitudinea lor, dar operațiile propriu zise sunt transferate și executate de către IDRISI. Nu există nici o posibilitate de afișare a fișierelor rezultate în aplicația creată.

O altă posibilitate de modelare prin limbaj de programare extern este *inovaGIS*, sistem bazat tot pe tehnologia COM, prin care se pot accesa direct fișierele raster sau vector a sistemului IDRISI.

Folosind doar acest sistem, nu este necesară prezența IDRISI pe calculator, însă în acest caz, folosind doar funcțiile *inovaGIS*, integrate într-un limbaj de programare înalt, toate operațiile de analiză ar trebuie rescrise în programul nou creat. *inovaGIS* oferă însă posibilitatea de afișare a fișierelor raster sau vector, interogarea valorilor de pe suprafața imaginilor afișate, modificarea directă a valorilor din fișiere.

În concluzie *inovaGIS* și *IDRISI API* integrate împreună într-un limbaj de programare poate reprezenta o soluție modernă și viabilă pentru modelarea unor procese complexe.

2.2 ERDAS Imagine 8.4

Posibilități de modelare internă

- modelare grafică
- modelare pe bază de macrouri
- modelare bazată pe limbaje script

Posibilități de modelare cooperativă

- C-Toolkit + Visual C++

Posibilități de modelare externă

- nu există

Datorită faptului că această aplicație este puțin mai veche, posibilitățile de analiză în modelare spațială se axează mai mult pe varianta internă, nu există posibilitatea de modelare externă, întrucât în timpul dezvoltării versiunii 8.4 (1999) nu era răspândită încă ideea de cooperarea a programelor prin tehnologii COM, ActiveX sau OLE.

Model Maker-ul revoluționar din vremea respectivă utilizează elemente grafice interconectabile. Fiecare element grafic corespunde unui obiect care poate intervenii într-un model: fișiere de intrare și de ieșire (raster, vector, tabel), funcții de calcul numeric sau spațial etc. Există posibilitatea de ramificare a modelului prin introducerea unor funcții condiționale. Nu există posibilitatea de realizare a structurilor repetitive.

Modelul grafic realizat prin *Model Maker* poate fi transformat în limbaj script, numit și *SML* (Spatial Modeler Language), care are o ofertă mai bogată în structuri de control incluzând în această categorie atât deciziile cât și structurile repetitive. Modele *SML* pot fi realizate și direct fără ca o variantă a modelului să fie în prealabil creat prin *Model Maker*.

Ambele posibilități prezentate (*Model Maker*, *SML*) suferă în privința comunicării cu utilizatorul. Parametrizarea modelului și obținerea unor date de ieșire altele decât imagini raster se poate realiza foarte greu.

O altă posibilitate de modelare internă este *EML* (ERDAS Macro Language), este de fapt mai mult decât un limbaj macro. Este mai mult un limbaj prin care se pot realiza interfețe de lucru (ferestre de dialog) pentru modelele construite în *SML*. *EML* ține loc pentru un API (Application Programming Interface) generic, care în zilele noastre se află încorporat în orice sistem de operare.

Singura opțiune care depășește granița sistemului ERDAS Imagine, este un *Toolkit*, o bibliotecă de funcții integrabilă în Visual C++, care deși oferă control asupra datelor și permite utilizarea operațiilor de analiză, utilizarea lui este foarte greoaie.

2.3 ArcView 3.2

Posibilități de modelare internă

- modelare bazată pe limbaj script

Posibilități de modelare cooperativă

- AVPython+Python

Posibilități de modelare externă

- inovaGIS

Limbajul de programare intern în ArcView 3.2, *Avenue*, este un limbaj de programare puternic, modern, bazat pe obiecte, oferind o gamă largă de posibilități atât în privința manipulării cât și a analizei datelor. Este însă un limbaj închis, nepremițând crearea unor obiecte proprii. Numeroasele greșeli de programare în crearea sistemului ArcView, fac ca orice aplicație realizată în acest sistem să fie instabilă. Greșelile de programare sunt accentuate în momentul în care datele utilizate nu sunt sută la sută corecte și lipsite de erori.

Avenue poate comunica cu aplicații bazate pe Visual Basic prin DDE (Dynamic Data Exchange), tehnologie existentă și pusă la dispoziție de sistemele de operare Windows.

AVPython este o extensie ArcView care permite utilizarea programelor scrise în limbajul Python în ArcView. În același timp este valabilă și formulare inversă, adică în limbajul Python pot fi integrate aplicații ArcView. Prin această posibilitate se pot extinde clasele existente și se pot crea obiecte proprii.

Tehnologia *inovaGIS* este aplicabilă și în cazul ArcView, întru-cât formatul de fișiere accesibile prin această posibilitate include și formatul SHP, fișierele vector. Toate cele amintite la IDRISI Kilimanjaro Edition sunt valabile și în acest caz.

3. Studiu de caz

Pentru exemplificarea celor amintite s-a luat un exemplu fictiv, și anume: prin măsurări s-au determinat înălțimea unor puncte. Se dorește generarea unui model de înălțime după care se dorește selectarea și calculul de arie pentru acele zone care au o pantă și o expoziție a versanților specificată.

Pentru exemplificare s-a ales IDRISI Kilimanjaro Edition întru-cât în acest caz sunt prezente toate cele trei posibilități de modelare (intern, cooperativ și extern).

Logica modelării presupune următorii pași:

- realizarea unui fișier vector pe baza punctelor măsurate
- interpolarea suprafeței pe baza fișierului vector
- filtrarea modelului de înălțime
- calcularea pantei și a expoziției
- selectarea zonelor care corespund criteriilor
- calcul de arie

3.1 Macro Modeler

Utilizând Macro Modeler parametrii fiecărei comenzi se pot alege în fereastra care apar la un click cu butonul drept al mouse-ului în dreptul dreptunghiului asociat. De remarcat faptul că nu toate posibilitățile comenzilor sunt accesibile prin această modalitate. De exemplu pentru interpolare, comanda specifică a sistemului permite definirea totală spațiului prin coordonate, dar folosind comanda în Macro Modeler acest lucru nu mai este permis, spațiul fiind limitat de punctele extreme din fișierul vector.

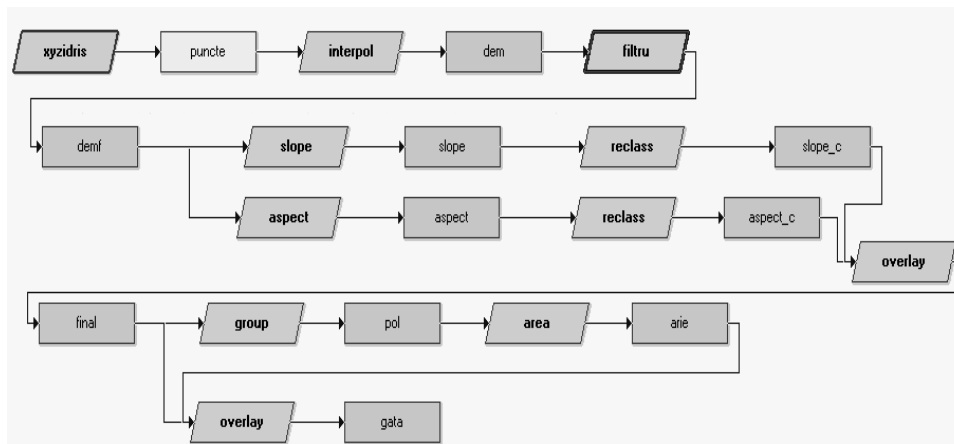


Figura 1. Schema de execuție în Macro Modeler

Un alt aspect este posibilitatea de modularizare prezent în acest exemplu prin submodelul denumit filtru.



Figura 2. Repetiție cu pași predefiniți în Macro Modeler

În acest caz se observă posibilitatea de repetiție a comenzilor, însă numărul de repetiții trebuie specificat numeric, fără să poată fi condiționată printr-o altă modalitate.

În momentul execuției modelului valorile dorite pentru pantă și expoziție trebuie înscrise în lista de parametri a comenzilor de calcul (SLOPE și ASPECT).

Cu toate neajunsurile sale Macro Modeler oferă o prezentare sugestivă a operațiilor înlănțuite și un control facil asupra parametrilor, împreună cu posibilitatea de vizualizarea a fișierelor rezultate.

3.2 Macro

```
rem TRANSFORMARE IN LAYER VECTORIAL
xyzidris x 1*1*puncte.txt*puncte*plane*m*1
```

```
rem INTERPOLARE SUPRAFATA
interpol x 1*puncte*dem*m*1*2*Y*2*140*140*w*fer.txt
```

```
rem FILTRE DE NETEZIRE SURAFATA
filter x dem*demf1*1
filter x demf1*demf2*1
filter x demf2*demf*1
```

```
rem CALCUL PANTA SI EXPOZITIE  
surface x 3*demf*slope*aspect*d*1  
  
rem CLASIFICARE PANTA  
reclass x i*slope*slope_c*2*0*0*%1*1*%1*%2*0*%2*90*-9999  
  
rem CLASIFICARE EXPOZITIE  
reclass x i*aspect*aspect_c*2*0*0*%3*1*%3*%4*0*%4*360*-9999  
  
rem SELECTARE PROPRIETATI COMUNE  
overlay x 3*slope_c*aspect_c*final
```

Figura 3. Secvență de macrouri care realizează analiza

Folosind fișierul macro se pot beneficia de toate posibilitățile comenzilor, fără nici o îngrădire, însă trebuie renunțat la iterația din Macro Modeler.

Există în plus și posibilitatea de parametrizare a modelului, astfel modelul este descris general și doar în momentul execuției anumite variabile primesc valori prin parametri. Parametrii în cadrul comenzilor macro sunt notate cu %1, %2, numărul după % reprezentând numărul de ordine a parametrului.

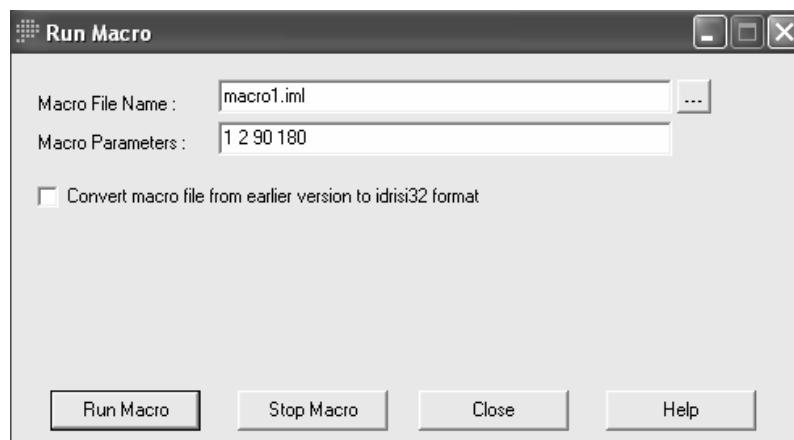


Figura 4. Lansarea în execuție a unui macro parametrizat

3.3 IDRISI API + Delphi

Această variantă implică un limbaj de programare de nivel înalt, cum ar fi Borland Delphi. IDRISI trebuie înregistrat în acest mediu de programare ca server de aplicație după care vor fi disponibile toate comenzile din IDRISI inclusiv acelea care definesc starea sistemului.

Folosind această posibilitate s-a realizat o aplicație în Delphi, cu o interfață de comunicare cu utilizatorul pentru a putea gestiona analiza problemei enunțate.

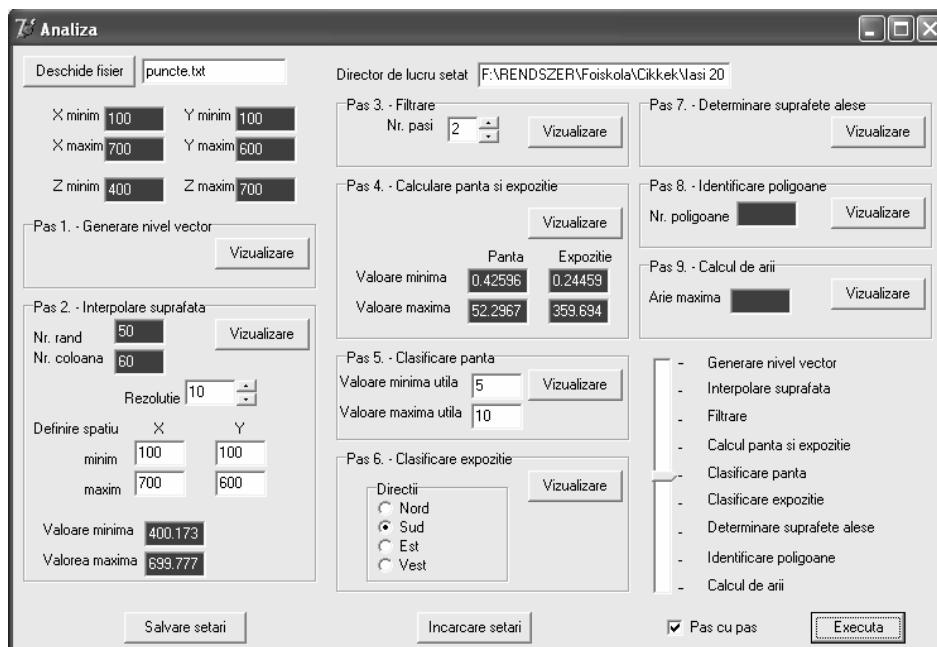


Figura 5. Interfața aplicației prin care se poate rezolva analiza

O astfel de interfață prezintă următoarele avantaje:

- definire interactivă a parametrilor (ex. rezoluția , limite de spațiu, număr de repetiție pentru filtrul de netezire a suprafeței, valorile care definesc intervalul pentru panta, definire faculă pentru expoziție prin direcții și nu prin valori)
- observarea valorilor rezultate după setarea parametrilor (ex. valori minime și maxime pentru datele din fișierul conținând punctele măsurate, numărul de rânduri și de coloane bazate pe limita spațiului și a rezoluției, valori minime și maxime pentru înălțime și pantă)
- controlul reexecutării analizei prin bara de defilare care conține pașii de execuție a analizei, ne fiind necesară pornirea analizei de fiecare dată de la început.

Totuși această variantă suferă de un neajuns, vizualizarea fișierelor raster rezultate se poate realiza doar din mediul IDRISI, ceea ce înseamnă că utilizatorul va trebuie să comute permanent între cele două aplicații pentru a ține sub observație toată analiza.

3.4 IDRISI API + inovaGIS + Delphi

Incluzând și inovaGIS în mediul Borland Delphi (într-un mod asemănător ca și IDRISI API) aplicația prezentată la punctul anterior poate fi dezvoltată cu un modul cu ajutorul căruia se pot vizualiza și interoga fișierele raster rezultate.

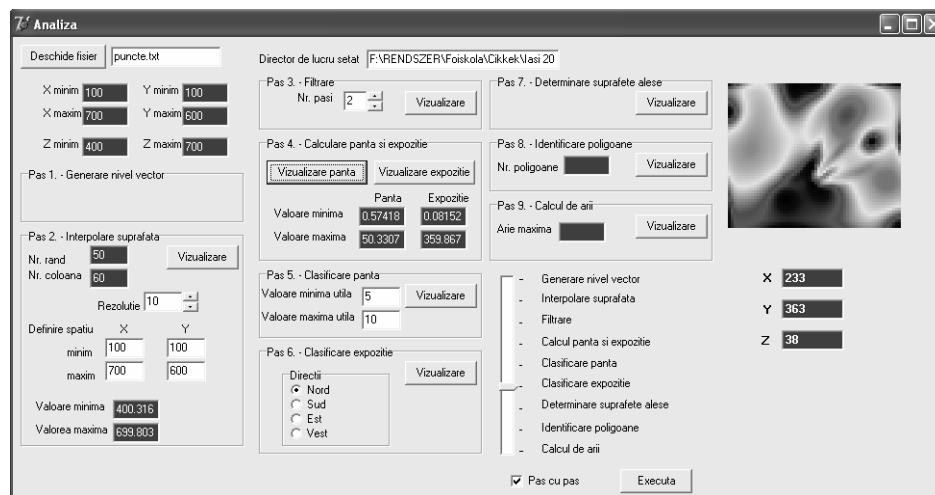


Figura 6. Interfața aplicației prin care se poate rezolva analiza cu vizualizarea fișierelor raster intermediare

4. Concluzii

Pentru rezolvarea unei modelări spațiale complexe deseori este necesară realizarea unui model parametrizat care conține decizii și/sau repetiții. Sistemele GIS rareori oferă posibilitatea de a construi un model care respectă aceste cerințe.

Combinarea unui sistem GIS cu un mediu de programare de nivel înalt poate avea ca rezultat un mini sistem GIS care satisface elementele specifice unei categorii de analize. Trebuie totodată luat în considerare că, modelele spațiale simple nu necesită dezvoltarea unei astfel de sistem.

BIBLIOGRAFIE

1. Brown, P., Butle, H. (2000) – *Beyond Avenue – Using Python with ArcView 3.x*, Iowa State University
2. Haidu I. (1998) – *S.I.G. Analiză Spațială*, Editura HGA, București
3. Mitasova H. (1998) – *Process Modeling and Simulations*, <http://www.ncgia.ucsb.edu/giscc/units/u130/u130.html>
4. Pereira, G.P. (2001) – *Help on inovaGIS Library*
5. *** (2003) – *IDRISI Applications Programming Interface (2003)*, User’s Guide, Clark Labs
6. *** (1997) – *ERDAS Imagine On-line Help*
7. *** (1998) – *Using the ERDAS Imagine Spatial Modeler*, GEO/EVS 425/525 Unit 8
8. *** (2000) – *Programming in ArcView 3.x GIS using Avenue*, Information Technology Service, Guide 94, Version 1.4